Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently amended) A method for source decoding a variable-length soft-input codewords sequence (y [1: T]) into a soft-output bit sequence (Λ_v [1: T]),

the variable-length soft-input input codewords sequence (y [1: T]) encoded in accordance with a VLC codewords table,

~~characterized in that it comprises~~the method comprising the acts of:

A. a first stage ~~(400)~~ of implementing a stack decoding algorithm for a sequential estimation of ~~an~~ a hard-output bit sequence of said variable length soft-input codewords sequence, wherein at each step of the sequential estimation, a stack of paths is reordered placing a current path having a smallest cumulative metric among paths listed in the stack at a top of the stack, including storage of intermediate data contained in the stack and generated by the stack decoding algorithm ; and

B. a second subsequent stage ~~(402)~~ of post-processing the stored intermediate data for generating the soft-output bit sequence (Λ_v [1: T]), a soft-output (Λ(x [t])) being provided for each bit.

2. (Currently amended) ~~Method~~ The method according to claim 1, ~~characterized in that~~ wherein the first stage ~~(400)~~ of implementing the stack decoding algorithm comprises the ~~steps~~ acts of:

- creating ~~(411)~~ ~~an~~ a unitary tree associated with said VLC codewords table, said unitary tree comprising nodes linked by branches, a path being defined by the branches from the initial node to each node of the tree;

- implementing the following sub-steps from an initial node of the unitary tree by using ~~a~~ the stack of paths, the stack having a current top path, each path being associated with a cumulative metric, the implementation being carried out until a set of stop conditions is verified:

- Computing ~~(113)~~ a metric M for each branch succeeding the current node of a current top path;
- If ~~(114)~~ the last node of the current top path corresponds to a codeword, concatenate the unitary tree with the current tree by placing the initial node of the unitary tree at least at the last node of the current top path;
- Deleting ~~(116)~~ the current top path from the stack;
- Inserting ~~(117)~~, in the stack, new extended paths made of the current top path and the succeeding branches, the cumulative metric of the new extended path being computed for each extended paths as the metric of current top path increased by the metric of the associated succeeding branch; and
- Selecting ~~(118) a~~ the new current top path according to the cumulative metrics associated to the paths.

3. (Currently amended) ~~Method~~ The method according to claim 2, ~~characterized in that~~ wherein the metric associated to a branch leading to a node l at time t is defined as follows:

m(l, y[t]) = - log P(y[t] | v(l)) - logpt(l) + log P0(y[t])

- where Np : the set of nodes having a predecessor ;
- pt(l) (l ∈ Np): the a priori probability of the branch reaching the node l at time t;
- v(l) (l ≡ Np): the value of the branch reaching the node l, v(l) ∈ {0,1};
- P0(y [t]) : a Fano - Massey metric which allows to compare fairly sequences of different lengths.

4. (Canceled)

5. (Currently amended) ~~A~~ The method according to any one of claims ~~2–4~~ 2-3, ~~characterized in that~~ wherein said set of stop conditions comprises the fact that the current top path contains the number of bits and the number of codewords of the variable-length soft-input codewords sequence (y [1: T]).

6. (Currently amended) ~~A~~ The method according to any one of the preceding claims ~~2-5~~, ~~characterized in that~~wherein the second subsequent stage ~~(102)~~ of post-processing the stored intermediate data comprises the step of approximating each soft-output Λ(x [t]) for each bit by:

$$\Lambda(x[t]) = \mu(t, 0) - \mu(t, 1)$$

where μ(t, 1) is the minimum cumulative metric for all the paths in the stack for which the t estimated bit is 1 and μ(t, 0) is the minimum cumulative metric for all the paths in the stack for which the t estimated bit is 0.

7. (Currently amended) A method ~~according to any one of the preceding claims 1-5, characterized in that~~ for source decoding a variable-length soft-input codewords sequence (y [1: T]) into a soft-output bit sequence (Λv [1: T]),

the variable-length soft-input input codewords sequence (y [1: T]) encoded in accordance with a VLC codewords table,

the method comprising the acts of:

A. a first stage of implementing a stack decoding algorithm for a sequential estimation of a hard-output bit sequence of said variable length soft-input codewords sequence, including storage of intermediate data contained in the stack and generated by the stack decoding algorithm ; and

B. a second subsequent stage of post-processing the stored intermediate data for generating the soft-output bit sequence (Λv [1: T]), a soft-output (Λ(x [t])) being provided for each bit, wherein the second subsequent stage ~~(102)~~ of post-processing the stored intermediate data comprises the step of approximating each soft-output Λ(x [t]) for each bit by:

$$\Lambda(x[t]) = \log\left(\sum_{\substack{1\le i\le r \\ T_{Pi}>=t \\ \hat{x}_{Pi}[t]=1}} e^{-\mu P_i} \Big/ \sum_{\substack{1\le i\le r \\ T_{Pi}>=t \\ \hat{x}_{Pi}[t]=0}} e^{-\mu P_i} \right)$$

~~Where~~ where Pi (i ∈ [1,...r]) are the r examined paths stored in the stack and μPi is the cumulative metric of path Pi, T_{Pi} is the length of path Pi and $\hat{x}pi$ (t) is the t^th hard bit of an hard bit sequence corresponding to path Pi.

8. (Currently amended) A computer program product for a decoder, comprising a set of instructions, which, when loaded into said decoder, ~~causes~~configure the decoder to carry out each of the ~~method claimed~~acts in any one of claims ~~1–7~~1-3.

9. (Currently amended) A computer product for a computer, comprising a set of instructions, which, when loaded into said computer, ~~causes~~configure the computer to carry out each of the ~~method claimed~~acts in any one of claim ~~1–7~~1-3.

10. (Currently amended) A decoder for source decoding a variable-length soft-input codewords sequence (y [1: T]) into a soft-output bit sequence (Λ_v [1: T]),

the variable-length soft-input input codewords sequence (y [1: T]) encoded in accordance with a VLC codewords table,

~~characterized in that it comprises~~the decoder comprising:

means for implementing a stack decoding algorithm for a sequential estimation of an hard-output bit sequence of said variable length soft-input input codewords sequence, wherein at each step of the sequential estimation, a stack of paths is reordered placing a current path having a smallest cumulative metric among paths listed in the stack at a top of the stack, including storage means for storing intermediate data contained in the stack and generated by the stack decoding algorithm ; and

means for post-processing the stored intermediate data for generating the soft-output bit sequence (Λ_v [1 : T]), a soft-output (Λ(x [t])) being provided for each bit.

11. (Currently amended) A Method for iterative decoding a variable-length soft-input sequence (z [1: Tx n/k]) of Tx n/k bits comprising the following ~~steps~~acts repeated for each iteration r:

- computing a variable-length soft-output bit sequence ($\tilde{\Lambda}_C^{(r)}$ [1 : T]) by applying a channel decoding method using as input the variable-length soft-input sequence (z [1: Tx n/k]) and a priori probabilities ratio (ϕ (r - 1) [1: T]) calculated for the previous iteration (r – 1);

- computing a variable-length soft-input codewords sequence ($y^{(r)}$ [t]) depending on the variable-length soft-output bit sequence ($\widetilde{\Lambda}_{C}^{(r)}$[1: T]) provided by applying the channel decoding method;

- computing a variable-length soft-output bit sequence ($\Lambda_{V}^{(r)}$[t]) by applying a method for source decoding according to any one of claims ~~1-7~~1-3 using as input the variable-length soft-input codewords sequence (y(r)[t]) depending on the variable-length soft-output bit sequence ($\widetilde{\Lambda}_{C}^{(r)}$ [1: T]) provided by applying the channel decoding method ; and

- computing the a priori probabilities ratio (ϕ (r) [1: T]) depending on the variable-length soft-input bit sequence ($\Lambda_{V}^{(r)}$ [t]) applied by said method for source decoding.

12. (Currently amended) A receiver for iterative decoding a variable-length soft-input sequence (z [1: Tx n/k]) comprising means for implemented the following ~~steps~~ acts repeated for each iteration r:

- computing a variable-length soft-output bit sequence ($\widetilde{\Lambda}_{C}^{(r)}$ [1 : T]) by applying a channel decoding method using as input the variable-length soft-input sequence (z [1: Tx n/k]) and a priori probabilities ratio (ϕ (r - 1) [1: T]) calculated for the previous iteration (r – 1);

- computing a variable-length soft-input codewords sequence (y(r) [t]) depending on the variable-length soft-output bit sequence ($\widetilde{\Lambda}_{C}^{(r)}$ [1: T]) provided by applying the channel decoding method;

- computing a variable-length soft-output bit sequence ($\Lambda_{V}^{(r)}$[t]) by applying a method for source decoding according to any one of claims ~~1-7~~1-3 using as input the variable-length soft-input codewords sequence y(r)[t] depending on the variable-length soft-output bit sequence ($\widetilde{\Lambda}_{C}^{(r)}$ [1: T]) provided by applying the channel decoding method ; a

- computing the a priori probabilities ratio (ϕ (r) [1: T]) depending on the variable-length soft-input bit sequence ($\Lambda_{V}^{(r)}$ [t]) applied by said method for source decoding.

13. (New) The method according to claim 1, wherein a position of each path listed in the stack other than the current path having a smallest cumulative metric is maintained when the stack is reordered.

14. (New) The method according to claim 1, wherein if the last node of the current top path corresponds to a codeword, the method comprising the act of concatenating the unitary tree with the current tree by placing the initial node of the unitary tree at least at the last node of the current top path.

15. (New) A program stored on a computer readable medium arranged for source decoding a variable-length soft-input codewords sequence (y [1: T]) into a soft-output bit sequence (Λ_v [1: T]), the variable-length soft-input input codewords sequence (y [1: T]) encoded in accordance with a VLC codewords table, the apparatus comprising:

a program portion configured to implement a stack decoding algorithm for a sequential estimation of a hard-output bit sequence of said variable length soft-input codewords sequence, wherein at each step of the sequential estimation, a stack of paths is reordered placing a current path having a smallest cumulative metric among paths listed in the stack at a top of the stack, including storage of intermediate data contained in the stack and generated by the stack decoding algorithm ; and

a program portion configured to post-process the stored intermediate data for generating the soft-output bit sequence (Λ_v [1: T]), a soft-output (Λ(x [t])) being provided for each bit.

16. (New) The program of claim 15, wherein a position of each path listed in the stack other than the current path having a smallest cumulative metric is maintained when the stack is reordered.

17. (New) The program of claim 15, wherein if the last node of the current top path corresponds to a codeword, the program comprising a program portion configured to concatenate the unitary tree

with the current tree by placing the initial node of the unitary tree at least at the last node of the current top path.